



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/016,619	12/06/2001	Stephen C. Netemeyer	PM 2000.062	6577

7590 01/10/2006

Gary D. Lawson
ExxonMobil Upstream Research Company
P.O. Box 2189
Houston, TX 77252-2189

EXAMINER

PROCTOR, JASON SCOTT

ART UNIT	PAPER NUMBER
----------	--------------

2123

DATE MAILED: 01/10/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/016,619	Applicant(s) NETEMEYER ET AL.	
	Examiner Jason Proctor	Art Unit 2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 20 October 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13 and 16-22 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-13 and 16-22 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 06 December 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>see continuation</u> . | 6) <input type="checkbox"/> Other: _____ |

IDS statements:

6/27/02

9/30/05

10/20/05

12/9/05

DETAILED ACTION

Claims 1-16 were rejected in Office Action of 18 July 2005. Applicants' response dated 20 October 2005 has amended claims 7, 10, and 13; cancelled claims 14-15; and presented new claims 16-22.

Claims 1-13 and 16-22 have been rejected.

Specification

The Examiner thanks Applicants for amending the specification regarding the use of trademarks.

The Examiner concurs that no new matter has been added.

Information Disclosure Statement

The Examiner thanks Applicants for resubmitting the Information Disclosure Statement that was originally filed on 26 February 2002. The references listed have been considered.

Claim Rejections - 35 USC § 101

35 U.S.C. § 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

1. Claims 10-13 and 16-22 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

MPEP 2106 reads as follows:

The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation or research (*Brenner v. Manson*, 383 U.S. 519, 528-36, 148 USPQ 689, 693-96); *In re Ziegler*, 992, F.2d 1197, 1200-03, 26 USPQ2d 1600, 1603-06 (Fed. Cir. 1993)). Accordingly, a complete disclosure should contain some indication of the practical application for the claimed invention, i.e., why the applicant believes the claimed invention is useful.

Claim 10 recites a method that fails to produce a useful, concrete, and tangible result.

The step of "using the mathematical simulation of transport phenomena to manage the facility network" is a statement of intended use and lacks a positively recited link between "the mathematical simulation" and "manag[ing] the facility network". The claim has a clearly stated utility, however this is insufficient to define a statutory method under 35 U.S.C. § 101. The claim language, as amended, fails to produce a result.

Additionally, the Examiner respectfully submits that though the preamble refers to a "computer system," none of the recited method steps directly require or employ the computer system. Therefore it would be improper to interpret the amended limitation as somehow meaning that the computer system is integral to managing the facility network.

The previous Office Action included a suggestion that the claim should "achieve some physical transformation outside of the computer, such as controlling the facility network, as appropriate." The Examiner's intention was not so broad as to suggest that a statement of intended use would meet the requirements set forth in MPEP 2106. To overcome this rejection, the Examiner respectfully suggests claim language that completes the link between "a mathematical simulation" and "manag[ing] the facility network." For example, this could mean making adjustments to the facility network so that its operation corresponds to a desired result from the mathematical simulation. However, care should be taken that the preamble corresponds

Art Unit: 2123

with the result of the method, i.e. a method for “simulating” results in “simulation” whereas a method for “controlling” results in an active step of “controlling”.

Claim 13 is rejected for similar reasons. Claim 16 is rejected for similar reasons corresponding to the limitation “using the simulation to manage reservoir and a delivery location of transport phenomena of the hydrocarbon facility network.” Claims rejected but not specifically mentioned stand rejected by virtue of their dependence.

To expedite a complete examination of the instant application the claims rejected under 35 U.S.C. § 101 (nonstatutory) above are further rejected as set forth below in anticipation of applicant amending these claims to place them within the four statutory categories of invention.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 1-9 and 13 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

MPEP 2164.01(a) lists several factors that can contribute to the determination whether experimentation is “undue”. These factors include, but are not limited to:

- The state of the prior art;
- The nature of the invention;
- The level of one of ordinary skill;
- The existence of working examples; and
- The quantity of experimentation needed to make or use the invention based on the content of the disclosure.

The limitation of “the extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself” prevents a person of ordinary skill in the art from making and using the invention. The state of the prior art is such that existing programming languages that support object-oriented programming by various implementations define a class hierarchy according to the class types. The class hierarchy is a representation of the existing object types. The addition of new object types to the hierarchy, through inheritance, derivation, or some other equivalent, necessarily changes that class hierarchy. Evidence of working examples of programming languages that exhibit the claimed behavior would be appreciated.

The nature of the invention appears to be a means for simulating transport phenomena such as extracting oil from a subterranean hydrocarbon-bearing reservoir, not directly related to programming language design. In order to make and use the invention as claimed, a person of ordinary skill in the art would be required to invent an undisclosed programming language that facilitates this particular behavior, an endeavor unrelated to “simulating transport phenomena”.

The act of designing computer programming languages is highly complex, undertaken almost exclusively by highly educated experts in that specific field, and often requires a prohibitive investment in time and money. Programming languages such as Ada have gone through numerous versions throughout the years, and can be referred to as Ada95 (1995), Ada83 (1983), and so on.

As a result of these considerations, undue experimentation would be required to make and use the claimed invention wherein “the extensible class hierarchy” would permit “the addition of additional object types and additional member variables without any modifications to the class hierarchy itself”.

Claims rejected but not specifically mentioned stand rejected by virtue of their dependence. Claim 13 reiterates the limitation discussed above.

In response, Applicants’ argue primarily that:

While Applicants respectfully submit that the Examiner has not satisfied the requirements set forth within the M.P.E.P. § 2164.01(a) for establishing an enablement rejection, Applications have included citations to at least some of the passages of the specification that provide support for the claimed subject matter based on the discussion with the Examiner during the telephonic interview. [...] When working with the system, a user may create one or more facility instances of the facility types and attribute values of attribute types to model a simulation. *See id* at page 16, line 32 to page 17, line 5. Thus, the present technique provides a mechanism for permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself. *See id* at page 15, line 26 to page 17, line 17.

The Examiner respectfully traverses this argument as follows.

The Examiner thanks Applicants for citing support for the claimed subject matter found in the specification. The Examiner has carefully considered these portions of the specification. However, these passages provide substantially the same explanation of the limitations at issue as presented in Applicants’ arguments, shown above. To demonstrate, page 15, lines 27-31 states:

For the computer system of the present invention, a novel approach has been taken to separate much of the member variables from the objects to which they belong. The class hierarchy encompasses two sub-hierarchies of classes, one sub-hierarchy containing classes to define a variety of "facility types" and the other sub-hierarchy containing classes to define a variety of "attribute values."

This teaching is in accordance with prior art object-oriented programming languages, however this does not provide enabling support for an "extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself" because this portion of the specification merely describes a prior art programming language that does not exhibit the property referred to in the limitation.

Page 16, lines 4-10:

Each attribute value object contains a specialized unit of data that helps to define the characteristics of the facility object. Conceptually, the facility is defined by a collection of associated objects--the facility object itself, plus all of its associated attribute value objects. This class architecture does not strictly follow the conventional OOP practice of encapsulation, because a facility's data is not contained fully within a single object.

This teaching reveals that an object (*facility*) does not correspond to a single software object, yet there still exists a marked deficiency regarding enabling support for an "extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself." From all appearances, creating an "additional object type" as required by the claim, regardless of the OOP practice of encapsulation, modifies the class hierarchy.

Page 16, lines 11-15:

A further unique characteristic of the computer system of the present invention is that both the facility type classes and attribute value classes are "generic." Specifically, this means that the most specialized classes in each of the two previously described sub-hierarchies are capable of representing many different, more specialized facility types, and many different attribute values. For example, one type of specialized facility class is Node (203, FIG. 2). This class is generic because Node objects can be instantiated to represent a variety of different node facility types, even though the class hierarchy does not have a distinct specialized node class for each node facility type. Likewise, the SystemAttributeValue (212, FIG. 2) is a generic attribute class because SystemAttributeValue objects can be instantiated to represent a variety of different attribute types, such as floating point scalars, integer scalars, strings, enumerated types, floating point arrays, and integer arrays.

Art Unit: 2123

This passage appears to directly address the limitation at issue, but neither this nor the surrounding text explains how the feature in the limitation may be performed with existing programming languages. Instantiating generic classes with specialized data is extremely well known in programming languages and one of the explicit goals of “constructor” methods, however constructor methods do not achieve an “extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself.” Merely changing or assigning data within a class or object does not and cannot meet the known-in-the-art definition of “creating a new object type”. These concepts are distinct and unrelated.

Page 16, lines 23-28:

A key advantage of the class hierarchy containing generic facility type and attribute value classes is that the collection of facility types and attribute values supported in the software system can be modified (additional facility types and attribute values defined, or existing ones removed) with no change to the class hierarchy, thereby avoiding the need to recompile source code to accommodate changes to facility types or attribute values.

This passage suggests the limitation at issue but again fails to disclose enabling support. How can additional object types be added to the hierarchy or existing object types be removed without changing that hierarchy? A given hierarchy is defined in terms of the existence and arrangement of its members. Applicants’ claim limitation requires changing a hierarchy without changing that hierarchy, which is a logical impossibility.

The Examiner reiterates the request for evidence of working examples of programming languages that exhibit this claimed behavior. Applicants have thus far declined to show what existing programming languages would enable the invention, but provide further evidence supporting the Examiner’s analysis that conventional, prior art programming languages are

Art Unit: 2123

insufficient to enable Applicants' invention. Applicants' response, page 12 of 20, referring to the rejection under 35 U.S.C. §102:

However, Stroustrup does not describe an "extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself," as recited in claim 1. Indeed, the modification of the emergency class and vehicle class, as described in Stroustrup, does change the class hierarchy. *See id.* at page 735. As such, Stroustrup fails to disclose the claimed subject matter of claim 1.

It is worth noting that Bjarne Stroustrup is the creator of the ubiquitous C++ programming language. The Examiner respectfully submits that Stroustrup is somewhat of an authority regarding the capabilities of the C++ programming language. Applicants' analysis of Stroustrup seems accurate. Stroustrup teaches a conventional, known in the art method of extending a class hierarchy by creating new object types. Clearly, creating new object types in C++ does change the class hierarchy.

It remains unknown how Applicants' invention manages to provide an "extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself." The Examiner maintains that achieving this limitation would require at least the creation of an undisclosed programming language. Applicants have provided no credible alternative explanation in either the remarks or in the disclosure of the application. Applicants' arguments have been fully considered, but have been found unpersuasive.

The following is a quotation of the second paragraph of 35 U.S.C. § 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

The previous rejection of claim 7 under 35 U.S.C. § 112, second paragraph, has been withdrawn.

3. Claims 10-12 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 10 makes reference to “simulating transport phenomena in a facility network” but also recites, “building a model comprising a facility network” and “wherein the facility network comprises facility instances ... and member variable instances.” As a result, it is unclear if a “facility network” is a tangible network of, for example, pipes and pumps, as seemingly disclosed by the specification, through which “transport phenomena” naturally occur; a component of a model that may or may not be a computer-based model; or comprised of “facility instances” and “member variable instances” that appear to be software engineering concepts. Numerous interpretation problems arise with each of these definitions. It is unknown what “transport phenomena in a facility network” means where a “facility network” is software or a component of a model. It is unclear how a non-computer-based model comprising a “facility network” can comprise software engineering concepts. Given the broadest reasonable interpretation of the claim language in light of the specification, the language of claim 10 presents contradicting implicit definitions of the phrase “facility network”.

Claims 11 and 12 stand rejected by virtue of their dependence.

4. Claim 18 is rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 18 recites “wherein the facility types comprises **one or more of** surface flowlines, manifolds... **and any combination thereof**” which renders the scope of the claim indefinite. The language is at least redundant and it is unclear whether the inclusion of the phrase “and any combination thereof” changes the scope of the claim. The claim will be interpreted without the phrase “and any combination thereof.” Clarification is respectfully requested.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1 and 8-9 are rejected under 35 U.S.C. § 102(b) as being anticipated by “The C++ Programming Language, Third Edition” by Bjarne Stroustrup (1997).

Regarding claim 1, Stroustrup discloses a computer programming language that implicitly discloses the use of a modern computer system comprising memory means, storage means, and software created using the C++ programming language.

Stroustrup discloses a class hierarchy (page 735, either figure) comprising a first set of generic classes representing a plurality of object types (example: *Car* or *Truck*) and a second set of generic classes representing member variables for the object types [“*The ‘plain’ cars and trucks are initialized with `Vehicle::eptr` zero; the others are initialized with `Vehicle::eptr`*”

Art Unit: 2123

nonzero.” (page 736) Also, example: *Police_car*, class definition of *Police_car*] and wherein the hierarchy is designed to be expanded as necessary [section 24.3.2.1 Dependencies within a Class Hierarchy; “*If, however, the intent is to provide a framework into which a later programmer can add code, then virtual functions are often an elegant mechanism for achieving this...*” (page 738)].

The use of the object-oriented extensible class hierarchy for the storage of transport phenomena simulation data is regarded as intended use. Stroustrup discloses the use of the extensible class hierarchy for the storage of vehicle data (page 738). MPEP 2111.02 reads as follows:

If a prior art structure is capable of performing the intended use as recited in the preamble, then it meets the claim. See, e.g., *In re Schreiber*, 128 F.3d 1473, 1477, 44 USPQ2d 1429, 1431 (Fed. Cir. 1997) (anticipation rejection affirmed based on Board’s factual finding that the reference dispenser (a spout disclosed as useful for purposes such as dispensing oil from an oil can) would be capable of dispensing popcorn in the manner set forth in appellant’s claim 1 (a dispensing top for dispensing popcorn in a specified manner)) and cases cited therein. See also MPEP § 2112 - § 2112.02.

Although the intended use is not recited by the preamble, the body of the claim is directed toward the structure of the extensible class hierarchy and thus defines the intended use for that hierarchy. The class hierarchy of Stroustrup is clearly capable of performing the intended use as stated, for example, by replacing the vehicle data with transport phenomena simulation data, and therefore meets the claim.

In response, Applicants’ argue primarily that:

However, Stroustrup does not describe an “extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself,” as recited in claim 1. Indeed, the modification of the emergency class and vehicle class, as described in Stroustrup, does change the class hierarchy. See *id.* at page 735. As such, Stroustrup fails to disclose the claimed subject matter of claim 1.

The Examiner respectfully traverses this argument as follows.

Applicants' arguments are unpersuasive for primarily two reasons. The first reason is that the limitation referred to in Applicants' arguments is not enabled by the disclosure and appears to claim a logical impossibility. However, according to MPEP 2143.03, limitations that do not have support in the specification as originally filed must still be considered. The Examiner has done so by presuming that Applicants do not intend to claim a method including a logical impossibility and has interpreted the claim accordingly. As no credible alternative interpretation for the claim language has been presented, the limitation referred to by Applicants is interpreted according to the mechanics of known and operable programming languages. As noted by Applicants, Stroustrup discloses the mechanics of a known and operable programming language. The rejection is therefore proper.

Secondly, Applicants' attention is respectfully drawn to the particular language of the claim, specifically the use of the word "permitting." Limitations using language such as "enables," "is capable of," or "permitting" do not require the predicate feature to be shown, but rather that the predicate feature *is not prohibited*. Although Stroustrup discloses the mechanics of a known and operable programming language, Stroustrup does not explicitly prohibit an "extensible class hierarchy permitting the addition of additional object types and additional member variables without any modifications to the class hierarchy itself." Therefore, regardless of the support in the specification for such a limitation, Stroustrup indeed anticipates the language of the claim.

Applicants' arguments have been fully considered but have been found unpersuasive.

Regarding claim 8, Stroustrup discloses software written in C++ (pages 732-733).

Regarding claim 9, Stroustrup discloses a hierarchy of logically related data (page 735, either figure, corresponding implementation illustrated on page 736).

The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, provides the following definition relied upon for this rejection:

database A collection of logically related data stored together in one or more computerized files.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 2-5 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Stroustrup as applied to claim 1 above, and further in view of US Patent No. 6,038,389 to Rahon et al. (Rahon).

Stroustrup does not expressly teach the details of Applicants' particular intended use, although the class hierarchy of Stroustrup is capable of performing the intended use of claims 2 through 4.

Rahon teaches a method of modeling a physical process in a material environment (abstract) with an exemplary use of a hydrocarbon reservoir. Rahon models the transport phenomena [“Pumping has been simulated during 86,400 seconds (1 day), with a flow rate of 30

Art Unit: 2123

m³/day.” (column 6, lines 48-50); “*FIG. 1* [a graph of kPa at t(s) (pressure dependent on time)] *compares the values obtained by means of the method according to the invention to the values obtained by means of a conventional time division type method...*” (column 6, lines 51-64)].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine Rahon’s method of modeling a physical process, such as the transport phenomena in a hydrocarbon reservoir through a well to the earth’s surface, with the class hierarchy capable of storing data as taught by Stroustrup. The combination could be achieved by representing the various components of Rahon’s model [*grid pattern, well, pumping, etc.* (column 6, lines 42-50)] as the objects (storing the appropriate simulation data) and methods of a class hierarchy as taught by Stroustrup. Motivation to do so is explicitly taught by Stroustrup, such as an enhanced ability to understand how the model works [“*The point about modeling reality is not to slavishly follow what we see but rather to use it as a starting point for design, a source of inspiration, and an anchor to hold on to when the intangible nature of software threatens to overcome our ability to understand our programs.*” (page 734)].

Regarding claims 3 and 4, Rahon teaches a transport pathway including a well and, by modeling pumping, implicitly teaches modeling a pump (column 6, lines 40-50). The combination and motivation to combine are the same as those for claim 2. A pump and a well constitute a facility network through which hydrocarbon fluids are transported between the subsurface reservoir and the delivery locations.

Regarding claim 5, the specification defines a “Data Definitions File”:

Art Unit: 2123

In the current system, clear-text file, for example, an ASCII file, which can be referred to as a "Data Definitions File" contains text entries that define each facility type and all of the attribute values for each facility type. (page 16, lines 28-31)

Stroustrup teaches a clear-text file that contains text entries that define each object type and all of the attribute values for the objects (excerpts shown on page 736, 737, et cetera), which are generally referred to in the art of software engineering as "class definitions" in "source code" (see pages 33-34).

In response to the rejections of claims 2-5, Applicants' argue primarily that:

Rahon does not cure the deficiencies of Stroustrup. [...] Rahon appears only to describe modeling of a subsurface reservoir, not the transport of hydrocarbons from the subsurface reservoir to the surface. [...] Indeed, the passages cited by the Examiner in Stroustrup do not even mention modeling of hydrocarbons, much less, a hydrocarbon reservoir. Similarly, the Rahon reference does not mention object-oriented programming much less object-oriented extensible class hierarchy. As such, neither reference suggests or teaches a motivation for the proposed combination.

The Examiner respectfully traverses this argument as follows.

Regarding Applicants' first point, the alleged deficiencies of the Stroustrup reference have been addressed above.

Regarding Applicants' second point, the Examiner respectfully submits that Applicants' interpretation of the Rahon reference is unduly narrow. The subject matter contemplated by Rahon encompasses transport of hydrocarbons from the subsurface reservoir to the surface, although perhaps not in Applicants' preferred vocabulary [*"Characterization of a petroleum reservoir is a delicate problem faced by geophysicists, geologists and reservoir engineers. When one tries to clarify or to validate the description of a geologic model, interpretation of well tests, of interference tests and of production results plays a very important part."* (column 1, lines 15-20, emphasis added); *"Pumping has been simulated during 86,400 seconds (1 day), with a flow rate of 30 m³/day."* (column 6, lines 48-50, emphasis added)].

Further, Applicants' arguments refer to limitations that define the properties of data in a method claim where the claim does not positively recite how those properties relate to the method. The "transport phenomena" could be defined as anything, but absent a positive recitation of how that definition changes the method, these limitations *do not further define the method*. As indicated in the body of the rejection, the Rahon reference has been cited to show that the field of Applicants' claimed intended use is known in the art.

Regarding Applicants' final point, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The Examiner has cited support in Stroustrup for using "software programs to model reality," as aptly described by Applicants. As illustrated above, these claims relate to an intended use for the method of claim 1, in particular, to use the method (*software program*) of claim 1 to simulate transport phenomena (*model reality*).

Applicants' arguments have been fully considered but have been found unpersuasive.

7. Claim 6 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Stroustrup as applied to claim 1 above, and further in view of US Patent No. 6,842,725 to Sarda.

Regarding claim 6, Stroustrup does not expressly teach the graphical user interface.

Sarda teaches a method for modeling fluid flows in a hydrocarbon reservoir (column 2, lines 5-18). Sarda teaches a user interface wherein a user defines the specific well network for the reservoir [detailed description, section 5, "Simulation input data", "*The data relative to the*

Art Unit: 2123

well are: its geometry, in the form of a series of connected segments[, and] the imposed flow rates, in the form of a curve giving the imposed flow rate as a function of time." (column 8, lines 28; 53-59)]. Sarda teaches a graphical interface concerning the well [*"A well is a series of connected segments that intersect the network fractures. The geometric representation of a well is therefore a 3D broken line."* (column 5, lines 60-63)]. Sarda both suggests and implies the use of a graphical user interface for defining the specific network of wells and facility objects to simulate transport phenomena into and out of a specific hydrocarbon-bearing reservoir.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Sarda regarding graphical modeling of fluid flows, specifically including a model of a well as a series of connected segments, with the object oriented software design taught by Stroustrup. The combination could be achieved by representing the various components of Sarda's model (segments of the well, nodes of the mesh, et cetera) as objects in a class hierarchy. Motivation to do so is explicitly taught by Stroustrup, such as an enhanced ability to understand how the model works [*"The point about modeling reality is not to slavishly follow what we see but rather to use it as a starting point for design, a source of inspiration, and an anchor to hold on to when the intangible nature of software threatens to overcome our ability to understand our programs."* (page 734)].

In response, Applicants' reiterate the alleged deficiencies of the Stroustrup reference which have been addressed above.

8. Claim 7 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Stroustrup as applied to claim 1 above, and further in view of “Design Patterns: Elements of Reusable Object-Oriented Software” by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (Vlissides, 1995).

Regarding claim 7, Stroustrup does not expressly teach a graphical user interface through which a user can define additional data members.

Vlissides teaches a design pattern for object-oriented software called the “Factory Method” (page 107) wherein an exemplary use is shown (page 107) depicting, in flowchart form, a user interface wherein a user of a computer system can create additional data objects (*Documents*) by using a graphical user interface. Vlissides shows an exemplary “user customized” data member [*MyApplication*] that contains its own user-customized *CreateDocument()* member function.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine the Factory Method design pattern taught by Vlissides, especially in light of Vlissides’ examples, with the object-oriented programming language taught by Stroustrup to achieve a graphical user interface through which a user can define additional data members. The combination could be achieved by using a user-customized *CreateFacility()* function, wherein the nature of the problem to be solved would motivate a person of ordinary skill in the art to customize that function to the particular intended use at hand. Motivation to combine is expressly taught by Vlissides [“*Use the Factory Method pattern when: a class can’t anticipate the class of objects it must create; a class wants its subclasses to specify the objects it creates; classes delegate responsibility to one of several helper subclasses, and you want to*”

Art Unit: 2123

localize the knowledge of which helper subclass is the delegate." (page 108) All these reasons relate directly to customization of the classes at a later point in time.].

In response, Applicants' reiterate the alleged deficiencies of the Stroustrup reference which have been addressed above.

9. Claims 10-12 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sarda in view of Stroustrup.

Regarding claim 10, Sarda teaches a method for modeling fluid flows in a hydrocarbon reservoir (column 2, lines 5-18) including a facility network [*"A well is a series of connected segments that intersect the network fractures. The geometric representation of a well is therefore a 3D broken line."* (column 5, lines 60-63)]. Sarda teaches a user interface wherein user specifies values of the member variables for each facility [detailed description, section 5, "Simulation input data", *"The data relative to the well are: its geometry, in the form of a series of connected segments[, and] the imposed flow rates, in the form of a curve giving the imposed flow rate as a function of time."* (column 8, lines 28; 53-59)]. Sarda teaches using the specified values in a mathematical simulation of transport phenomena within the facility network as a function of time [*"In order to simulate a well test, whatever the medium, this equation has to be solved in space and in time. Discretization of the reservoir (mesh pattern) is therefore performed and solution of the problem consists in finding the pressures of the meshes with time, itself discretized in a certain number of time intervals."* (column 2, lines 24-47)].

Sarda does not explicitly disclose the software design of the method for modeling fluid flows.

Stroustrup discloses “building a model comprising a facility network [...] formed from facility types based on a first set of generic classes and member variable instances formed from member variables for the facility types based on a second set of generic classes, and wherein the first set and second set of generic classes are part of a class hierarchy that is not modified by the addition of other facility types and member variables,” (pages 735-738 and as cited above). This limitation refers to basic concepts of object-oriented programming. Whether or not a class hierarchy is modified at a later time is impossible to compare to the teachings of the prior art.

It would have been obvious to a person of ordinary skill in the art to implement the model taught by Sarda in object-oriented software because the advantages of object-oriented software are well known to persons of ordinary skill. Motivation to do so is explicitly taught by Stroustrup, such as an enhanced ability to understand how the model works [*“The point about modeling reality is not to slavishly follow what we see but rather to use it as a starting point for design, a source of inspiration, and an anchor to hold on to when the intangible nature of software threatens to overcome our ability to understand our programs.”* (page 734)].

Regarding claim 11, Sarda teaches that the facility network is part of a larger simulation model, with said facility network being capable of exchanging fluids with at least one other part of the simulation model [fluid from the reservoir is transferred via the series of connected segments that form the facility network of the well (column 2, lines 55-67)].

Regarding claim 12, Sarda teaches that the simulation model comprises a facility network [*“A well is a series of connected segments that intersect the network fractures. The geometric representation of a well is therefore a 3D broken line.”* (column 5, lines 60-63)] and a hydrocarbon-bearing formation [*“Discretization of the reservoir (mesh pattern) is therefore performed and solution of the problem consists in finding the pressures of the meshes with time, itself discretized in a certain number of time intervals.”* (column 2, lines 43-47)].

Applicants' arguments regarding the rejection of claims 10-12 are moot in light of the new grounds of rejection.

10. Claim 13 is rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 6,434,435 to Tubel et al. (Tubel) in view of Sarda, and further in view of Stroustrup.

Regarding claim 13, Tubel teaches an object-oriented software for control of a hydrocarbon production system (column 1, lines 14-50). Although Tubel is primarily concerned with the control of such a system, the disclosed invention can also perform in a simulation mode [*“...in the simulation mode, simulated and/or calculated sensor and actuator data may be used in place of data from real-world sensors and actuators. Simulation of real or abstract systems occurs by having an ISO 10 evaluate or interrogate a model of a real or abstract thing or system or evaluate and/or interact with rules associated with the real or abstract thing.”* (column 12, lines 6-18)].

Tubel teaches a controlling (and therefore simulating) a hydrocarbon-bearing reservoir penetrated by a plurality of wells and surface facilities connected to the wells [*"...the present invention relates to management of hydrocarbon production from a single production well (e.g., only well 642) or from a group of wells, shown in FIG. 28 as well 640, well 641, and well 642."* (column 23, lines 8-15); *"Referring still to FIG. 28, as is well known in the art a given well may be divided into a plurality of separate zones, such as zone 640a, zone 640b, and zone 640c. Such zones may be positioned in a single vertical well such as well 640 associated with surface platform 645, or such zones may result when multiple wells are linked or otherwise joined together (not shown in FIG. 28)."* (column 26, lines 52-58, emphasis added)].

Tubel teaches using objects and variables in a class hierarchy to model the wells and surface facilities [*"Referring now to FIG. 30, a diagrammatic representation of ISOs 10 in flow and hierarchical relationships, ISOs 10 can model and represent any device or group of devices including sensors 200, controllable devices 300, fluid processing devices 400, injection devices 500, or any combination thereof. ISOs 10 can also model and represent more abstract processes such as a single zone like 640a, a group of zones such as 640 a and 640b, an entire well such as well 640, or an entire field such as wells 640, 641, and 642."* (column 25, lines 23-31)].

Tubel does not teach discretizing the reservoir into a plurality of volumetric cells, each modeled as nodes, and simulating the exchange of fluid between those nodes.

Sarda teaches discretizing the reservoir into a mesh pattern (consisting of interconnected nodes) used to model the reservoir by finding the pressure of the oil contained therein as a function of time (column 2, lines 24-47). In this method, the model simulates the flow of fluid through a porous medium, accounting for the real geometry of the fracture network (found in the

Art Unit: 2123

reservoir) and thus simulates the interactions between the pressure and flow rate variations in a well running across the medium (column 2, lines 55-67). Sarda teaches specifying initial conditions for each node and connection (column 4, lines 33-64).

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine Sarda's method for modeling the flow of fluid in a reservoir with Tubel's object-oriented method of modeling a facility network. The combination could be achieved by operating Tubel's method in simulation mode, wherein the data calculating the transport phenomena of the underground reservoir is supplied by Sarda's method and delivered to the sensors and actuators modeled by Tubel. Motivation to combine would be found in the knowledge of a person of ordinary skill in the art as well as the nature of the problem to be solved; Tubel's simulation mode requires calculated sensor and actuator data while the results of Sarda's model provide an efficient and accurate representation of the transport phenomena that would be detected by Tubel's sensors.

Tubel in view of Sarda does not explicitly teach the claimed software object hierarchy.

Stroustrup discloses a computer programming language that implicitly discloses the use of a modern computer system comprising memory means, storage means, and software created using the C++ programming language.

Stroustrup discloses a class hierarchy (page 735, either figure) comprising a first set of generic classes representing a plurality of object types (example: *Car* or *Truck*) and a second set of generic classes representing member variables for the object types [*"The 'plain' cars and trucks are initialized with Vehicle::eptr zero; the others are initialized with Vehicle::eptr nonzero."* (page 736) Also, example: *Police_car*, class definition of *Police_car*] and wherein

Art Unit: 2123

the hierarchy is designed to be expanded as necessary [section 24.3.2.1 Dependencies within a Class Hierarchy; *"If, however, the intent is to provide a framework into which a later programmer can add code, then virtual functions are often an elegant mechanism for achieving this..."* (page 738)].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine Tubel in view of Sarda with the class hierarchy capable of storing data as taught by Stroustrup. The combination could be achieved by representing the volumetric cells of Tubel in view of Sarda as the objects (storing the appropriate simulation data) and methods of a class hierarchy as taught by Stroustrup. Motivation to do so is explicitly taught by Stroustrup, such as an enhanced ability to understand how the model works [*"The point about modeling reality is not to slavishly follow what we see but rather to use it as a starting point for design, a source of inspiration, and an anchor to hold on to when the intangible nature of software threatens to overcome our ability to understand our programs."* (page 734)].

Applicants' arguments regarding the rejection of claim 13 are moot in light of the new grounds of rejection.

11. Claims 16-22 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 5,331,579 to Maguire, Jr. et al. (Maguire) in view of US Patent No. 6,002,985 to Stephenson.

Regarding claim 16, Maguire teaches a computer implemented method of modeling a power plant (column 2, lines 61-63) comprising computer software wherein:

The model is composed of a plurality of generic classes in a hierarchy [*"During the model initialization process the supervisor 50... initiates the plant object."* *"In this way initiation of the object tree is controlled from the top down while actual execution is from the bottom up."* (column 5, line 41 – column 6, line 35)];

The objects in the system are made up of a collection of other objects in the hierarchy [*"To add objects to the system, it is only necessary to add the new object to the objects list contained in the parent object... For example, a subcomponent object may be representing the bearings in a feedwater pump while the component object represents the pump which includes not only the bearings but a drive motor component."* (column 6, lines 18-35)];

The objects do not modify the class hierarchy [inherent, (column 6, lines 18-35)];

Simulating the model [*"The next step in the simulation process is to run the modeling system based on initial conditions and expected operating conditions for a period of time..."* (column 7, lines 12-16)]; and

Using the simulation to manage the power plant [*"The function of the system is to collect, store, and display data representative of the operating condition of the plant components and system. The system then calculates the expected life of each component and each system that includes the components. The system can also make recommendations directing the plant operator to perform or to refrain from performing certain procedures."* (column 3, lines 20-27)].

Maguire's disclosure is directed toward a power plant rather than a hydrocarbon system.

Stephenson teaches a method of controlling development of an oil or gas reservoir (column 2, lines 25-27), the reservoir including a plurality of wells drill therein from which oil or gas has been produced (column 2, lines 28-30), identifying production parameters associated with the production of the oil or gas from the plurality of wells (column 2, lines 41-42), and generating an output from a neural network to predict performance (column 2, line 60 – column 3, line 11; et cetera). Stephenson teaches a computer model of an oil or gas reservoir (column 3, lines 12-14) and including data representing the wells and physical parameters of the wells (column 3, lines 14-25).

Stephenson teaches additional benefits of the method [*“In particular, it is a computer-implemented method of controlling development of an oil or gas reservoir by enabling an individual to observe through the operation of the computer a simulated production of oil or gas from the reservoir before an actual well is drilled in the reservoir to obtain therefrom actual production corresponding to the simulation production.”* (column 4, lines 25-42)].

It would have been obvious to a person of ordinary skill in the art at the time of Applicants’ invention to combine the method of Maguire for modeling a system using hierarchical objects with the disclosure of Stephenson, teaching modeling an oil reservoir, as motivated by the teachings of the prior art [*“The present invention is a computer-based modeling system designed to improve the overall performance of components and systems that degrade with age”* (Maguire, column 1, lines 10-12); a person of ordinary skill in the art would recognize a hydrocarbon system as a “system that degrades with age”] as well as the knowledge of persons of ordinary skill in the art. Maguire teaches all but the intended use; a person of ordinary skill in the art at the time of Applicants’ invention would immediately recognize that Maguire’s method

Art Unit: 2123

of modeling a power plant, including pumps (column 6, lines 28-32; column 16, lines 3-5; et cetera) is applicable to a hydrocarbon system. The combination could be achieved by implementing Maguire's method in the scope of a hydrocarbon system and defining the systems and components as appropriate.

Regarding claims 17 and 19, Stephenson teaches modeling fluid transport between a surface facility and a subsurface formation accessed by a plurality of wells [*"FIG. 1 shows that each of the wells 6 has been drilled by a suitable drilling process 12."* (column 4, lines 63-64); *"Examples of production parameters 22 pertinent to the present invention include but are not limited to the following: day and year of start production, six month cumulative gas and/or oil production, and twelve month cumulative gas an/or oil production."* (column 6, lines 8-13)].

Regarding claim 18, Maguire teaches modeling a pump (column 16, lines 28-32; column 16, lines 3-5).

Regarding claim 20, Maguire teaches that the method is written in FORTRAN (column 5, lines 42-58).

Regarding claim 21, Maguire teaches using a text file configured to define the objects for use in the simulation, wherein the objects may be accessed without being coded as part of the application [*"To add objects to the system, it is only necessary to add the new object to the objects list contained in the parent object."* (column 6, lines 18-21); *"Result data produced by*

Art Unit: 2123

an object such as object 64 and 66 is stored in the common data pool 68 where it can be accessed by any object at any level, thereby providing data communication between objects.” (column 6, lines 22-28); *“Prior or subsequent to the error check 141, the controller preferably sets an indicator in the common database 68 that indicates that the controller module for this object has started a simulation cycle.”* (column 9, lines 11-15)].

Regarding claim 22, Maguire teaches creating new objects for represent components for simulation [*“To add objects to the system, it is only necessary to add the new object to the objects list contained in the parent object.”* (column 6, lines 18-21)].

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Art Unit: 2123


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

jsp


Paul L. Rodriguez 1/6/05
Primary Examiner
Art Unit 2125